Control architectures for Home Area Networks

Richard J. Bradbury

August 9, 1996

Abstract

The role of communications is already well established in the office environment, thanks to networking innovations such as Ethernet LANs. With the advent of cheap, affordable broadband communications and the increasing complexity of consumer goods, it seems natural to extend the network into the home. This document proposes a programme of research into such Home Area Networks and the control architectures required to support useful applications on them.

1 Introduction

The purpose of the Warren project is to investigate the application of high speed networking in the home environment. The particular area of interest described here is that of control in such a network. It is proposed to conduct research into three architectural aspects of control, namely:

- **Control protocols.** The actual network protocols used to control physical devices in the network.
- **Control software.** The software architectures used to drive the network protocols. It is anticipated that distributed object technologies will play a major role here.
- **User control.** The software and hardware required to make the devices in the network perform meaningful and useful functions. This will involve schemes for user interaction with the control system.

Because the Warren project concerns itself with new concepts in networking, there is no single existing platform on which these ideas can be properly tested. Instead, it is proposed that a Home Area Network be built from scratch. This provides an ideal opportunity to mix and match some of the best elements of current network thinking with the new ideas generated by the project.

Some work has already been done in defining a suitable network architecture [1], and is summarised in section 2. The architecture is largely derived from the bottom up and consequently reflects the practical needs of the project rather than idealistic concepts. The wisdom of such an approach will become more apparent with time.

2 Warren network architecture

The architecture of the Warren is that of a mesh of interconnected ATM switches (figure 1). In the terminology of the ATM Forum, the Warren would be considered a private logical network operating its own proprietary protocols. Full description of the proposed network architecture can be found elsewhere [1]; some interesting features are summarised below.

One of the key aspects in the project is to investigate the extent to which ATM can be built into low-cost consumer products. This is to be achieved by means of a number of economies:

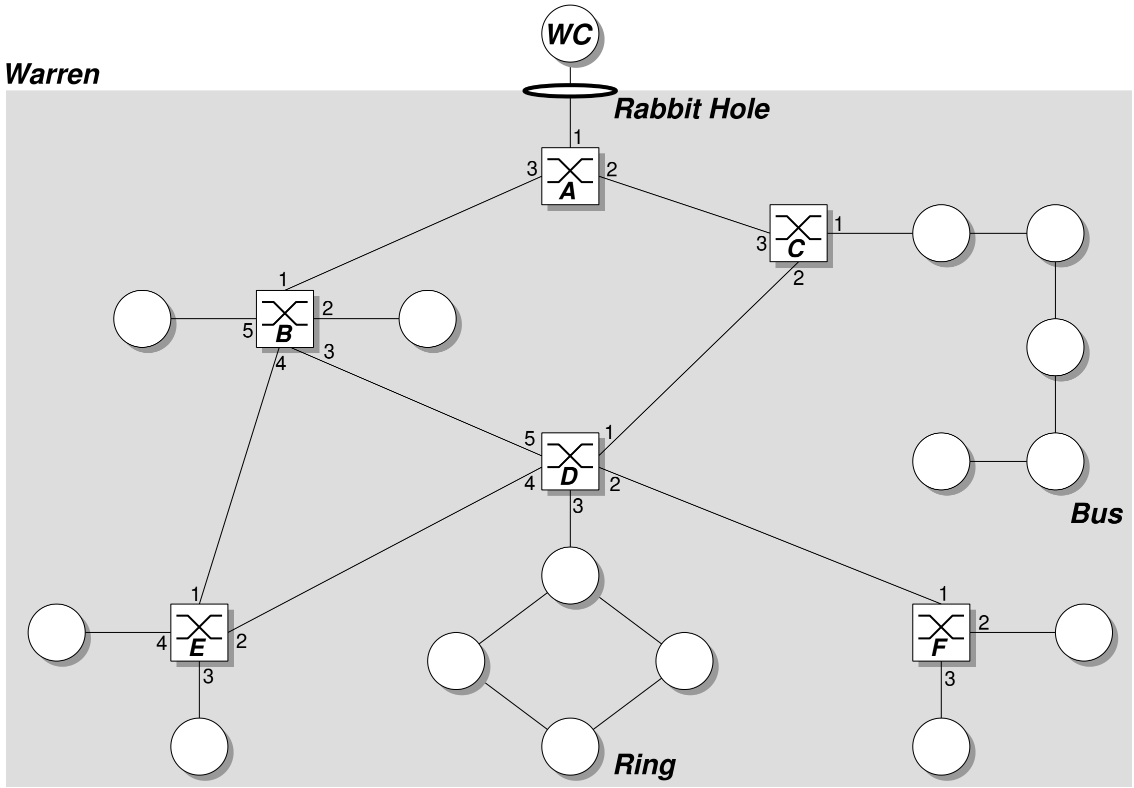

Figure 1: Example Warren network topology

- **Cheap transmission media.** The use of cheap twisted pair media will reduce cost. Line speeds of 25.6 Mbit/s will also provide economies in line terminating components.
- **Low cost switches.** ATM switches with small numbers of ports (typically three or four) will enable switches to be built much more cheaply. Again, the chosen line speed of 25.6 Mbit/s will keep cost down.
- **Common hardware components.** Using the same line termination designs in end stations and switches ought to produce beneficial economies of scale.
- **Novel topologies.** Certain devices which make poor use of the total bandwidth available on a single link can be connected together in a bus topology by means of a ‘thru’ port fitted to each device. All the devices in such a daisy chain are then able to share the bandwidth of a single switch port. This reduces the waste of bandwidth and the number of switches required.
- **Dumb devices.** Both switches and end stations are designed to be as dumb as possible. In particular, support for autonomous signalling is removed from the hardware and replaced by software proxy. End stations are designed to transmit and receive only on fixed VCI values. Simple control protocols, again on fixed VCIs, are built into devices to allow remote control by proxies.
- **No unique identifiers.** A novel proposal for the network architecture is that devices carry no unique identifiers, such as a 48-bit MAC address. Devices will instead be addressed by their position in the network and the control protocols will use source route addressing. The rationale behind this is that manufacturing costs are reduced if all devices of a certain type are identical. As a consequence of this, automatic topology determination schemes are necessary.

The decision to factor out intelligence from the Warren devices means that an external ‘brain’ is required to control the network. Each device will be controlled by a proxy software object. The intention is that these proxies may be distributed arbitrarily around the public ATM network, according to whatever management policies the owner desires. The intelligence of the Warren network itself is distributed amongst the switches, which may in turn be managed by proxy [2].

However, each Warren will possess a node on which privately managed proxies (e.g. switch proxies) can reside. This so-called *Warren Controller* will be a platform capable of supporting a distributed processing environment which hosts the proxy objects. As has been separately documented [3], the computational platform selected (in line with the low-cost policy) is a Linux PC and the DPE selected to run on it is ANSAware.

2.1 Benefits of Warren approach

The major deficiency in many of the existing industry standards for Home Area Networking is their limited scope. Although the control of appliances can be achieved through a single network, they do not cater for the transport of data. This limits their usefulness to the control of relatively simple devices. It is not practical, for example, to interconnect Hi-Fi and television equipment over all but the shortest distances because separate cabling is required to transport the audio and video signals in analogue form.

One goal of this research is to determine whether the integration of both control and data information into a single communications framework makes the problem of implementing Home Area Networks more tractable, or whether the additional complexity of such a unified scheme outweighs the potential benefits. In the proposed Warren architecture, all device control information and all data transport is carried over a single ATM mesh network.

The benefits derived from this include:

- **Single network.** The cost of installing a Home Area Network is reduced because all traffic passes through one set of network components. The utilisation of the components is also higher, making them more cost-effective. The disruption caused by the installation of such a network is also reduced.
- **Single wire.** Each device is connected to the network by a single drop cable. No additional wiring between devices is required. Devices with modest power requirements may also be able to draw this from a switch port by means of 'phantom powering' schemes.
- **Single connector.** Each device is attached to the network using an identical connector. This enables devices to be easily moved from one location to another. The software support for Warren devices implements a 'plug-and-play' scheme to avoid manual configuration of network components as much as possible.
- **Unitary security scheme.** Because the network is all-digital, security policies can be applied in a consistent manner. A single security architecture can be defined and the mechanisms it embodies need be implemented only once. Thereafter, the security policies selected for individual Warren applications can be policed and enforced consistently. Note that it is not a goal of this proposal to research security issues.

2.2 Comparison of the Warren approach with FireWire

One possible architecture for a Home Area Network is based on the IEEE 1394 ('FireWire') standard. Originally intended to replace interfaces such as RS-232, Centronics parallel and the SCSI bus as a general interconnect between computers and periferal devices, the FireWire bus operates at speeds of 100 Mbit/s, 200 Mbit/s or 400 Mbit/s. The standard is flexible enough to allow tree and branch wiring topologies to be employed. However, individual links are limited to a length of 4.6 m, which may prove restrictive in a general home or small office environment.

With IEEE 1394, each node in the tree requires a bridge which connects the bus hierarchy above to two or more output ports. This is analogous to a three-port ATM switch, and likewise represents the major cost in such a network.

The high bandwidth available on an IEEE 1394 bus would allow the transmission of uncompressed multimedia streams over the HAN. This has the advantage of divorcing the presentation technology (e.g. digital television set) from the compression format employed to deliver the data to the home over links of narrower bandwidth: a separate decompression node is added to reconstruct the video stream. This could be an MPEG 2 decoder today; when a superior compression technique is developed, the customer simply purchases a new decompression node which automatically works with his existing equipment. This model helps protect investment in consumer electronics equipment, whilst allowing the timely introduction of new technology at a modest incremental cost.

Of course, being a bus, the large bandwidth made available by FireWire is shared by all nodes. Using a switched ATM network yields a smaller bandwidth on individual links (25.6 Mbit/s, or perhaps 51.2 Mbit/s), but a potentially higher aggregate bandwidth across the whole Warren. Of course, the whole bandwidth is never visible to a single Warren end station, and so multimedia streams must be transported in compressed formats.

2.3 Network interoperability

Despite the perceived benefits of a single network architecture such as the Warren, it is both useful and desirable to be able to interoperate with other architectures. The initial target for Warren networking is necessarily higher value items such as Hi-Fi equipment, where the additional cost of an ATM line interface is small compared to the total cost of the device. For a small device such as a simple mains power controller, the cost of the ATM interface dominates and it may prove cheaper to simply interwork with an existing standard. This is particularly true of power line schemes, such as LonWorks or EHS, where existing mains wiring could be used.

At a physical level, interoperation requires the construction of ‘gateway’ devices which convert the physical signals from one network format to another. Data format conversion is also required; a scheme for encapsulating EHS messages in ATM cells must be defined, for example. In addition, there are logical differences in the operation of these other networks which require higher level protocol conversions. It is likely that software proxies will play a crucial role here.

3 Software support

The Warren devices are expected to be used in residential and office environments. The switches and end stations must therefore be designed with a non-technical user in mind. A notion particularly trendy in current PC products is that of ‘plug-and-play’ – the ability of devices to configure themselves automatically when plugged together.

From the point of view of the Warren project, this presents an interesting challenge: in order to make dumb devices interoperate seamlessly and automatically requires a not inconsiderable software development effort. Fortunately this problem can be attacked from a number of different angles, namely:

1. **Network configuration.** The mechanics of determining the topology of the network. Devices need to register themselves with the Warren Controller.
2. **Device typing.** Devices must be able to describe their own capabilities to the Warren Controller.
3. **Media typing.** A type system for different data streams will be required.
4. **Action typing.** A syntax and semantics of actions must be defined.
5. **Security.** Features to ensure that only authorised operations on Warren objects are permitted will need to be considered.

Clearly, there is substantial overlap between these areas (e.g. 1 with 2, 2 with 3). The solution to the problem must be found, therefore, by considering all these areas.

3.1 Device typing

Recall that Warren end-stations are designed, for reasons of simplicity and cheapness, to transmit and receive only on fixed VCI values. When describing the capabilities of a particular device, then, it is sufficient to list the active/sensitive VCI values along with the type of each channel (according to some media type system).

This is clearly the simplest scheme and would likely work well with data streams. However, when it comes to describing *control* streams, the solution begins to look less satisfactory. Many Warren devices are controlled by poking values into particular memory locations. The operation of placing a fixed value into some location could be captured by such a system, but to relate that to an operation such as "switch the washing machine on with the cotton programme" is beyond the scope of syntactical descriptions.

Another problem is the diversity of devices. For example, a washing machine is very different from an infra-red sensor. When it comes to proxy control of devices in the Warren Controller (or other remote proxy), an object manager is required for each device. The Controller needs to know the full syntax of device typing and will need the necessary software classes to control every possible Warren device. Every time the consumer purchases a new device, a corresponding software upgrade to his system will be necessary. Clearly, this flies in the face of the 'plug-and-play' paradigm.

The solution to this problem is not to embody the description of the device in a syntax, but to realise that the entity most capable of describing a device is the device itself. Every device could be designed with its own on-board software. In manufacturing terms, the addition of an 8 Kbyte ROM to the circuit board is very small, and since every unit of a particular model is identical, useful economies of scale result.

Then, when a device registers itself with the Warren Controller, it is a simple procedure for it to upload this software into the Controller. This software is then executed alongside the other proxy controllers in the computer. In an object-oriented programming environment, the control of the device can be abstracted through the use of a well-defined interface. The core objects in the Warren Controller operate on the proxy object through its methods. The particular value poked into a particular location via a particular VCI is opaque to the invoking object.

3.2 Media typing

In a typical Warren, there will be large numbers of devices attached to the network, each capable of transmitting and/or receiving only a fixed set of data types. It is no good, for example, attempting to output infra-red keypad strokes on an ATM loudspeaker. Consequently, it is necessary for the data channels in the Warren to be typed in some way. There are two possible approaches to this:

- **'Connectionless'.** Every application layer data packet must carry a small header indicating the type of data explicitly. Systems such as the Olivetti *Medusa* project [5] and the IEEE 1394 ('FireWire') [4] standard have used typed data streams which follow this model.

  In FireWire, for example, a generic audio/video format ('HyperLynx') has been defined in the Common Isochronous Packet (CIP) Layer. A 64-bit CIP header is prepended to application data which includes a 6-bit Format Identifier and a 24-bit Format Dependent Field. The former indicates the general format of the data (e.g. MPEG 2, digital VCR, vendor-unique), while the latter indicates options within the general format (e.g. frame rate in MPEG 2 streams).

- **'Connection-oriented'.** The type of the stream is stated at connection setup time and is thereafter implicit. This scheme follows the ATM concept of connections more closely by moving data type negotiation out-of-band and into the connection negotiation phase.

Clearly, the connection-oriented method offers a more efficient solution in terms of bandwidth overheads. On the other hand, the connectionless solution would appear to offer greater flexibility in that the format of data can change during the lifetime of a single connection. In the context of the Warren, however, this feature is unlikely to be useful since the majority of devices will only be able to transmit a limited number of orthogonal data types on fixed VCI values. For these reasons, a connection-oriented approach has been selected as the preferred method of data stream typing.

4 User interaction

Although the network of Warren devices can be made to be largely self-configuring, the operational behaviour of the devices and the ways in which they interact to achieve specific results cannot be automated and must be configured manually by the user. For example, if the user wishes to use the Warren for a home security application, he can install motion detectors which will register themselves with the Warren Controller. He may also buy a piece of software which implements security applications, e.g. raising alarms on certain conditions. However, a small amount of manual configuration by the user is still required to specify which sensors are to be used and what alarms are to be raised.

Clearly, the arbitrary configuration of objects is a non-trivial task in the general case. It is proposed that a graphical form of user interaction be employed for this task. This assumes that the user has access to some sort of graphical display device suitable for this purpose, such as a set-top box or Network Computer. A dumb input/output device, such as an LCD tablet with touch-sensitive display could equally be used if a suitable display proxy were developed.

A general mechanism for representing Warren devices graphically and for creating and editing logical interactions between them is necessary. Such an application must be simple to use, since the target audience is likely to be non-technical.

Implementation of this graphical user interface represents interesting research in its own right. The distributed nature of the Warren control software indicates that a portable, platform-independent programming language may be appropriate in this case. This will enable the same software to execute on heterogeneous hardware platforms with different operating systems, and yet retain the same functionality from a single source code.

So-called Network Programming Languages, epitomised by the likes of *Tcl*, *TeleScript* and *Java*, are of great interest in this context. Java, an object-oriented language designed with secure operation over communications networks in mind, includes an Abstract Window Toolkit class library suitable for the implementation of Graphical User Interfaces. Investigating the possibility of using such languages in both client and server scenarios would also prove interesting.

5 Research Goals

The principal goals of the research are as follows. Firstly, the proxy control of network devices is to be researched. Although the distributed control of switches is an overlapping research area with another project [2], the proxy control of end stations constitutes a new research area.

Second, the Warren represents a feasibility study for the application of low-cost networking to the domestic environment. The research will determine the viability of bottom-of-the-range broadband ATM in transporting both control information and data around the home in a single unified network.

Finally, it is intended to study the ways in which distributed processing environments can be embedded in consumer products, and the extent to which user-friendly interfaces can be provided for them which allow non-technical users to perform high-level configuration of the devices in the network.

To this end, an investigation of suitable environments for supporting user interfaces will form a significant portion of the research. The goal here is to provide interfaces which can be scaled to operate over a wide variety of control devices, including heterogeneous hardware platforms and mobile devices, with an emphasis on ubiquitous infra-red communications.

6 Conclusion

The Warren is in part a technology demonstrator to show the viability of ATM in the domestic environment, but also provides an interesting vehicle for trying out new architectural ideas, both hardware and software. A software framework for the control of such a network provides interesting insights into many different areas of research: architectures, object-oriented and distributed environments and human–computer interaction.

It is hoped that the end-product of the research will be not only a better understanding of the issues involved in Home Area Networking, but also improved techniques in implementing practical control systems for them.

References

[1] Greaves, David J. and Bradbury, Richard J. *Low Cost ATM and the ATM Warren*. To be published 1996. http://www.cl.cam.ac.uk/Research/SRG/HAN/Warren/docs/warren_paper.ps.

[2] Roscoe, Timothy and Rooney, Shaun. *Overview of the DCAN project*. Systems Research Group ATM Document Collection 4 (The Green Book). University of Cambridge Computer Laboratory. July 1995.

[3] Bradbury, Richard J. *First Year Report*. July 1996.

[4] Institute of Electrical and Electronic Engineers. *High Performance Serial Bus*. IEEE Standards Document 1394. 1995.

[5] Wray, Stuart and Glauert Tim. *The Medusa Applications Environment*. Technical Report 94-3. Olivetti Research Laboratory, Cambridge. 1994.